

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	09/607,370	Confirmation No. 8705
Applicant	:	Reiner KRAFT et al.	
Filed	:	June 30, 2000	
TC/A.U.	:	2178	
Examiner	:	Kyle R. STORK	
Docket No.	:	ARC9-2000-0046-US1	
Customer No.	:	23334	

PRE-APPEAL BRIEF REQUEST FOR REVIEW

The following remarks are submitted to be considered along with the Appellant's notice of appeal. The references and/or combination of references cited by the Examiner do not describe, teach, nor suggest the presently claimed invention.

CLAIMS 3-13, 16-19, and 21-25 ARE PATENTABLE OVER THE CITED ART

The Examiner at the last paragraph beginning on page 2 and continuing to the top of page 3 of the Final Office Action states that Meyerzon teaches:

Executing, at the server, a web-browser, as part of the web crawler, wherein the web-browser displays an in-memory copy of the data document which has been retrieved, wherein the in-memory copy of the data document maintains a web-browser display format and a web-browser display layout of the dynamic data document when displayed in the web browser (Meyerzon Col 7 Lines 60-65 and Col 8 Lines 15-20: Here, the crawler acts as a web browser in that it requests the web page data. These requested web page documents are stored in memory in a display format)

Meyerzon does not teach a web-browser at the server as part of the web crawler. The Examiner states that "a crawler acts as a web browser in that it requests the web page data". However, the independent claims state that a web-browser is executed as part of the web-crawler. In other words, the web-crawler claimed in the present invention isn't merely acting as a web-browser but rather a separate web browser is being executed as part of the web crawler.

Furthermore, Meyerzon does not teach that a web-browser “displays an in-memory copy of the data document which has been retrieved, wherein the in-memory copy of the data document maintains a web-browser display format and a web-browser display layout of the dynamic data document when displayed in the web browser”. Merely retrieving web pages and parsing the pages to obtain text and formatting, as Meyerzon teaches, does not automatically mean that Meyerzon is teaching these claim elements. Meyerzon is merely parsing HTML code. For example, Meyerzon explicitly states that text and properties are obtained from tags within the HTML documents. See Meyerzon at column 9, lines 9-43. Therefore, Meyerzon is working on HTML source code. The information is passed to an indexing engine which creates an index of the retrieved data. The index contains reference information and pointers to corresponding electronic documents, for example, keywords. See Meyerzon at col. 8, lines 1-16. Therefore, Meyerzon cannot teach the above claim element.

The Examiner also states at the first full paragraph on page 3 of the Final Office Action that Meyerzon teaches:

Executing, at the server instead of a client system, a browser scripting engine as part of the web-browser for loading content as directed by the client-side scripting code into the in-memory copy creating a final web-browser display representation of the dynamic data document so that the final web-browser display representation is substantially similar to when the data document is viewed by a user in the user's web-browser running on the client system when all the data is viewed (Meyerzon Col 7 Lines 60-65 and Col 8 Lines 15-20)

Meyerzon is only teaching that a web crawler is retrieving electronic documents. Meyerzon is completely silent on “executing, at the server instead of a client system, a browser scripting engine as part of the web-browser for loading content as directed by the client-side scripting code...”. The Examiner has failed to show where in Meyerzon this claim element is at least suggested.

The Examiner correctly acknowledges that Meyerzon does not mention:

wherein the server processing unit renders the in-memory webpage prior to analyzing and summarizing the in-memory webpage.

However, the Examiner goes on to combine Meyerzon with Blumenthal stating that:

Blumenthal mentions a document that can be rendered prior to user actions (Blumenthal Col 17 Lines 45-53).

The cited portions of Blumenthal are limited to the additional rendering of a cached document at the **client** computer. In contrast, the presently claimed invention recites that the **server** “renders an in-memory copy of the dynamic data document which has been retrieved, wherein the in-memory copy of the dynamic data document maintains a rendered web-browser display format and a rendered web-browser display layout of the dynamic data document when the web-browser renders the in-memory copy of the dynamic data document. Furthermore, Blumenthal only teaches rendering additional areas of a cached document and not a complete in-memory webpage. See *Blumenthal* at col. 17, lines 45-52 and FIG. 13. The present invention, on the other hand, renders an in-memory webpage as it would be displayed on a user’s web browser and not just areas of the webpage. Therefore, the present invention is able to summarize the document based on the whole and complete document as it was designed by the document’s author; the static heterogeneous data, as well as the problematic dynamic data, is completely rendered and integrated into the metadata for subsequent indexing of all metadata by a web crawler.

On the top of page 4 of the Final Office Action, the Examiner also correctly acknowledges that Meyerzon does not disclose:

[...] wherein the data document is a dynamic data document, wherein an in-memory copy of a dynamic data document is rendered, and wherein a browser scripting engine executes the client-side scripting code.

However, the Examiner goes on to combine Meyerzon with Koike, stating that Koike discloses:

[...] a proxy server assembling a dynamic data document for display at a client browser wherein an in-memory copy of a dynamic data document is rendered, and wherein a browser scripting engine executes the client-side scripting code (Figures 6-8; column 7, lines 13-33).

The Appellants respectfully suggest that the Examiner has mischaracterized Kioke with respect to the presently claimed invention and has improperly used the Appellants' disclosure as a blueprint to reconstruct the claimed invention out of isolated teachings of the prior art.¹

Kioke merely teaches the dynamically generating a page in advance of a user page request and then storing the dynamically generated page. Generating a page is **not** the same as **rendering**. For example, column 7, lines 13-33 of Kioke states:

FIG. 6 also shows components involved in a batch page generation process. A batch page generation control unit 13 receives batch page generation definitions from a batch page generation definition area 20 and outputs a page generation signal to the page generation unit or program 12 in response to a time trigger signal from a timer unit 16 or a data update trigger signal from a data update unit 14. The time trigger signal is generated after a predetermined time period while the data update trigger signal is generated when the data update unit 14 updates the data in the data area 22 from the data table 31 via the DBMS 15. The above described batch page generation process prepares the updated dynamic WWW pages in advance of the user requests. This advance dynamic WWW page generation enables high-speed access by substantially eliminating on-the-fly page generation upon the user requests. Because of the advance dynamic WWW page generation, the preferred embodiment of the high-speed dynamic page generation system according to the current invention also takes advantage of a proxy server on the network.

In other words, Kioke teaches building a web prior to a user request for a page so that on-the-fly generation is reduced. Kioke further teaches in column 7, lines 64 to column 8, line 2, that once the page has been generated, a file name is determined so that the page can be later accessed. This clearly shows that Kioke is building a page and not rendering an in-memory copy of the dynamic data document. Nowhere does Kioke teach the rendering of these pages that are generated.

Furthermore, nowhere does Kioke teach or suggest “wherein the browser scripting engine executes the client-side scripting code and loads content as directed by the client-side scripting code into the in-memory copy...” Nowhere does Kioke teach a browser at the server executing a scripting engine, or that a scripting engine loads content as directed by client-side scripting code. Kioke merely teaches using data and a template to build a page. FIG. 8 clearly shows the

¹ See MPEP §2143 and *Grain Processing Corp. v. American Maize-Products*, 840 F.2d 902, 907, 5 USPQ2d 1788 1792 (Fed. Cir. 1988) and *In re Fitch*, 972 F.2d 160, 12 USPQ2d 1780, 1783-84 (Fed. Cir. 1992).

motivation behind Koike. For example, FIG. 8 shows that the system of Koike determines if a page requested by the user changes. If the page does change, the system of Koike builds the page so that the next time the user access the page, the user's system is not required to build the page. The Examiner even states in the Response To Arguments section of the Final Office Action that Koike **assembles** a dynamic document. As discussed above, assembling is not the same as "executing, at the server instead of a client system, a browser scripting engine as part of the web-browser, wherein the browser scripting engine executes the client-side scripting code and loads content as directed by the client-side scripting code into the in-memory copy creating a final web-browser display representation of the dynamic data document ...".

Accordingly, in view of the remarks and arguments given above, the combination of Kohl and Cherry fail to teach or suggest the presently claimed invention. Accordingly, claims 3-13, 16-19 and 21-25 recite in allowable form and the Examiner's rejection under 35 U.S.C. § 103(a) has been overcome and should be withdrawn.

The Appellants hereby respectfully request reconsideration and allowance of pending claims 3-13, 16-19 and 21-25 of the instant application.

Respectfully submitted,

Date: June 12, 2008

By: /Jon Gibbons/
Jon Gibbons
Attorney for the Applicants
(Reg. No. 37,333)

FLEIT, GIBBONS, GUTMAN
BONGINI & BIANCO P.L.
551 N.W. 77th Street, Suite 111
Boca Raton, FL 33487
Tel (561) 989-9811
Fax (561) 989-9812